

Försättsblad för skriftlig tentamen vid Linköpings Universitet

Sal	<i>SU15-16</i>
Tid	<i>08-12</i>
Kurskod	<i>TDP004</i>
Provkod	<i>DAT2</i>
Kursnamn/benämning	<i>Objekt-orienterad programmering</i>
Institution	<i>IDA</i>
Antal uppgifter som ingår i tentamen	<i>Teoretisk del: 6 Praktisk del: 2</i>
Antal sidor på tentamen (inkl. försättsbladet)	<i>Teoretisk del: 3 Praktisk del: 5</i>
Jour/Kursansvarig	<i>Per-Magnus Olsson</i>
Telefon under skrivtid	<i>1456</i>
Besöker salen ca kl.	<i>Ungefär varje hel timme</i>
Kursadministratör	<i>Anna Grabska Eklund</i>
(namn + tfnr + mailadress)	<i>anna.grabska eklund@liu.se</i>
Tillåtna hjälpmedel	<i>Teoretisk del: Inga Praktisk del: Den C++ information som finns i systemet. Lösningförslag kommer på kurshemsidan.</i>
Övrigt	
(exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)	
Vilken typ av papper ska användas, rutigt eller linjerat	<i>Spelar ingen roll.</i>
Antal exemplar i påsen	<i>6</i>

Tentamen i TDP004

Objektorienterad Programmering

Teoretisk del

- Datum: 2011-04-28
- Tid: 08-12
- Plats: SU-salar i B-huset.
- Jour: Per-Magnus Olsson, tel 281456
Jourhavande kommer att besöka skrivsalarna ungefär varje timme under skrivtiden.
- Hjälpmedel: Teoretisk del: Inga.
Praktisk del: Den C++ information som finns i systemet.
- Betygsättning: Max antal poäng: 42 med 21 poäng vardera på teori och praktikdel.
- | Poäng | Betyg |
|-------|----------|
| 36-42 | 5 |
| 29-35 | 4 |
| 22-28 | 3 |
| 0-21 | U |

Anvisningar: Börja med den teoretiska delen. När du är klar med den lämnar du in den och får den praktiska delen. När du har lämnat in den teoretiska delen kan du inte återvända till den.

Skriv svaret på varje teoretisk uppgift på ett separat blad.

Uppgifterna är inte ordnade efter svårighetsgrad.

Lycka till!

TDP004 Objektorienterad Programmering

Teoretisk del

1) En del av följande uppgifter innehåller felaktigheter. Rätta till de felaktiga, om du tycker att någon eller några är korrekta skriver du det. (5p).

- a) `int number = new int(3);`
- b) `cin << new_int;`
- c) `float decimal_number == 123.4;`
- d) `float &float_ref;`
- e) `double double_number = 5678;`

2) Eftersom arv är användbart, då måste väl multipelt arv vara mångdubbelt så användbart? I kursen tog vi upp ett potentiellt problem med multipelt arv. Beskriv detta och ge ett exempel på när det kan inträffa. (4p)

3) Vad används `const` till i C++-sammanhang? (2p)

4) Nedanstående kod kompilerar men kraschar när den exekveras. Förklara varför. (2p)

```
int & calculate_average(const std::vector<int>& v)
{
    std::vector<int>::const_iterator j;
    int sum = 0;
    for(j = v.begin(); j != v.end(); j++)
    {
        sum += (*j);
    }
    return sum;
}
```

5) I C kan man bara explicit typkonvertera på ett sätt. I C++ kan man däremot typkonvertera på inte mindre än fyra olika sätt.

- a. Ge namnen på dessa fyra. (2p)
- b. Förklara två (valfria) av dessa med var sitt exempel. (2p)

6) Virtuella funktioner kan vara mycket användbart i en del sammanhang.

- a. Vad är det som skiljer en virtuell funktion och en rent virtuell funktion? Du kan förutsätta att det är vanliga medlemsfunktioner som avses. (2p)
- b. Vad har dessa för effekt på klassen där funktionerna är definierade? (2p)